Grouping Stack Overflow Comments to Improve Software Development Practices

Devishree Jothilingam
Department of Computer Science, Lakehead University
Thunder Bay, Ontario, Canada
djothili@lakeheadu.ca

Haishwarya Sholingapuram Asuri Kannan
Department of Computer Science, Lakehead University
Thunder Bay, Ontario, Canada
hsholing@lakeheadu.ca

Aarthi Lakshmipathy
Department of Computer Science, Lakehead University
Thunder Bay, Ontario, Canada
alakshm4@lakeheadu.ca

Harshavardhan Subramanian Madhavan
Department of Computer Science, Lakehead University
Thunder Bay, Ontario, Canada
hsubram1@lakeheadu.ca

ABSTRACT

Developers today rely on Stack Overflow as a vital tool for interpersonal interaction, seeking assistance, and knowledge transfer. However, as the platform develops, users are presented with an overwhelming volume of comments, which can make it difficult for them to quickly locate pertinent information. The entire user experience is negatively impacted by this problem. In response, we propose a method that organizes comments based on similarities, allowing users swiftly identify common patterns while eliminating repetition and irrelevance. By eliminating unnecessary comments, this approach not only makes it less difficult to find relevant information but also helps to maintain the quality of the content. As a result, users are able to engage in more worthwhile discussions that enhance the platform's content. For the purpose of clustering, our suggested approach analyses the text content of comments using natural language processing (NLP) techniques. Preprocessing techniques on the comments comprise stemming, lemmatization, and tokenization. For the purpose of trying to group comments based on similarity, the pre-processed data is subsequently input into the K-Means clustering algorithm. This popular unsupervised learning approach works well for clustering and cosine similarity is used for recommending top 5 comments which are similar in context and content. This paper makes a substantial addition to the Stack Overflow community by addressing a pressing problem and offering a workable solution. By making navigation easier as well as rendering it less difficult to find relevant data, the recommended approach has the potential to significantly enhance the user experience. This approach additionally promotes discussions with higher caliber information by retaining content quality. Overall, our article offers a useful service to the Stack Overflow community and is transferable to other websites with comparable commenting systems.

Keywords: Stack Overflow, Comment classification, Comment clustering, Information retrieval, Data pre-processing, Feature extraction, Clustering algorithms, Content quality, Text similarity, Duplicate detection, Information filtering, Evaluation metrics.

INTRODUCTION

Stack Overflow is a well-known platform that has revolutionized the way programmers seek assistance and share knowledge worldwide. It is a popular community of more than 10 million registered users that has over 50 million monthly visitors. With its thriving community, Stack Overflow provides a platform for developers to collaborate on open-source projects and contribute to the development of various tools and libraries widely used in the industry.

However, the vast number of comments on Stack Overflow can make it challenging for users, especially new ones, to navigate and locate the relevant information they need. This issue has led to the proposal of an innovative solution that utilizes machine learning and natural language processing techniques to group comments based on their similarity, length, and user ID. This approach aims to simplify the process of locating relevant information and improve the overall user experience.

Before clustering, data preprocessing is necessary to prepare the comment data for analysis. The preprocessing stage involves cleaning the data, removing unnecessary information, and transforming the data into a format that can be easily analyzed by a machine learning algorithm.

The K-means clustering algorithm is an unsupervised machine-learning technique used to group similar data points together based on their characteristics. In this case, the algorithm is applied to Unix Stack Exchange comments and their associated metadata to uncover patterns and categories that may provide insights into user behavior and content quality [8].

The Adjusted Rand Index (ARI), Adjusted Mutual Information (AMI), and accuracy are metrics commonly used in evaluating the performance of clustering or classification models.

The Adjusted Rand Index measures the similarity between the predicted and actual clusters, taking into account chance agreement. It ranges from -1 to 1, where a score of 1 indicates perfect clustering agreement and a score of 0 indicates clustering that is no better than random. A negative score indicates clustering that is worse than random.

The Adjusted Mutual Information is a measure of the mutual information between the predicted and actual clusters, adjusted for chance. It ranges from 0 to 1, where a score of 1 indicates perfect clustering agreement and a score of 0 indicates clustering that is no better than random.

Accuracy is a measure of the proportion of correct predictions made by a classification model. It is calculated by dividing the number of correct predictions by the total number of predictions.

In summary, these metrics provide a quantitative measure of how well a clustering or classification model is performing. While accuracy measures the overall performance of a model, ARI and AMI are useful for evaluating the quality of clustering algorithms by taking into account chance agreement.

One way to enhance the user experience on Stack Overflow is by implementing a recommendation system that suggests the top 5 comments for each post. These comments can offer additional insights, solutions, or explanations to the original question or answer, ultimately assisting users in better understanding and resolving their programming issues [3].

The recommendation system operates by taking a post id and a comment id as input. Utilizing cosine similarity, it identifies the top 5 comments that are related to the given comment. The process is illustrated in the Recommendation system figure.

By employing this recommendation system based on cosine similarity, we can provide users with a selection of the most relevant and helpful comments for each post. This approach not only enhances the user experience on Stack Overflow but also contributes to improving the overall quality of the platform.

In conclusion, this project proposes a solution to enhance the user experience on Stack Overflow by utilizing machine learning and natural language processing techniques to group comments based on their similarity, length, and user ID. The solution encompasses the collection of comment data, preprocessing, feature extraction, comment clustering, the visualization of clustered comments via a user-friendly interface and recommend top 5 similar comments within a post and it's cluster. With its potential to streamline the process of locating relevant information and preserve content quality on the platform, the proposed solution could have a significant impact on the Stack Overflow community.

MOTIVATION

To ask and answer programming-related challenges, software developers frequently use the widely popular online forum Stack Overflow. The comment box on each question or answer on Stack Overflow is an essential component where users can offer opinions, clarifications, as well as additional details. However, users may find it difficult to identify and read through all relevant comments considering the enormous amount of comments posted each day. Additionally, it may be tricky for users to identify similar comments because they might be dispersed across several threads or pages. In order to promote user collaboration and communication, there is a critical need for a technique of grouping relevant comments that is more efficient and effective.

Our motivation behind this project is to address this issue by exploring advanced clustering algorithms for Stack Overflow comments. We aim to enhance the user experience by accurately grouping similar comments, making it easier for users to find relevant discussions and solutions. By incorporating natural language processing techniques and feature engineering for better comment representation, we strive to develop an effective clustering approach that overcomes the limitations of existing methods. The purpose of this project is to investigate cutting-edge clustering methods to analyze comments on Stack Overflow in a quest to solve this problem. By precisely categorizing pertinent comments, we aim that this will enhance the user experience by making it more simple for users to identify pertinent topics and solutions. We desire to develop an efficient clustering methodology that overcomes the constraints of existing methods by integrating natural language processing techniques and feature engineering for better comment representation. Our study aims to improve the effectiveness of knowledge discovery in online communities like Stack Overflow by analyzing the effectiveness of these algorithms in enhancing the user experience as well as recommending to users the most relevant comments to read.

Objective

The objective is to carry out text preprocessing, group comments using k-means clustering, analyze the results using several metrics, and showcase the results using a confusion matrix. The primary objective of the project is to cluster comments in a CSV file based on their preprocessed content, length, and user id. The performance of the clustering will then be assessed using a variety of metrics, and the findings will be visualized for additional analysis and decision-making.

1. **How does the practice of grouping comments in Stack Overflow contribute to more effective knowledge sharing and community engagement among developers?** Objective 1: In Stack Overflow, grouping comments is essential since it provides structured and organized discussions within a question's discussion area. Users may more easily follow and engage in pertinent discussions when comments are categorized under certain threads or themes, which improves knowledge-sharing and community participation among developers.
2. **How effective is the k-means clustering algorithm in clustering comments based on their preprocessed text features?** Objective 2: Using metrics like the Adjusted Rand Index (ARI), Adjusted Mutual Information (AMI), and accuracy, the research study is going to evaluate the

manner in which the k-means clustering algorithm clusters comments based on their preprocessed text characteristics.

3. **How accurate is the kmeans clustering result compared to the manual clustering result for a specific dataset?** Objective 3: The accuracy of the k-means clustering result when compared to the manual clustering result depends on a number of variables that are including the dataset's properties, the K value selection, the similarity measure employed, and the manual clustering expert's skill. The level of agreement between the clustering results and the ground truth (i.e., manual clustering) can be compared using clustering evaluation metrics such as the silhouette score, adjusted Rand index, or F-measure in order to determine the accuracy.

LITERATURE SURVEY

There have been several previous works that have addressed the issue of grouping and recommending Stack Overflow comments to improve software development practices. One such work is "Mining Stack Overflow to Turn the IDE into a Self-confident Programming Prompter," by Ponzanelli, Luca, et al. (2014)[2]. This work proposes a machine learning-based approach to mine and classify Stack Overflow comments and then use them to suggest code changes to developers while they are writing code. The approach uses a combination of natural language processing and machine learning techniques to extract relevant comments from Stack Overflow and then classifies them based on their relevance to a specific programming task.

Another study by Sheth, Viral, and Kostadin Damevski (2022) [6] propose a method to group and recommend Stack Overflow comments to software developers based on their relevance and similarity to the current development task. The proposed method uses a combination of semantic analysis and clustering techniques to group comments that are related to the task at hand. The authors also conducted experiments to evaluate the effectiveness of their approach and showed that it outperforms existing methods in terms of precision, recall, and F1-score. Overall, the study provides a novel approach to recommend relevant and useful comments from Stack Overflow, which can potentially improve the productivity and efficiency of software developers.

In a different study, Zhang, Haoxiang, et al.(2021) [11] analyzes the organization of comments on Stack Overflow, a popular Q&A platform for developers. The authors conduct a large-scale empirical study to investigate how well comments are organized for easy retrieval by developers. Their findings indicate that while Stack Overflow provides some useful organization features, there is still room for improvement in terms of organizing comments to facilitate easy retrieval by developers. The paper also provides several recommendations for improving the organization of comments on Stack Overflow.

Furthermore, Ranasinghe, Prasadhi, Nipuni Chandimali, and Chaman Wijesiriwardana. (2022) [4] investigate the characteristics of useful comments in Stack Overflow, a popular online forum for programmers to ask and answer technical questions. The authors conduct a systematic analysis of comments across multiple programming languages and identify various types of useful comments, such as those that provide explanations, corrections, or suggestions for improvement. They also develop a classification system for useful comments and demonstrate its effectiveness in predicting the usefulness of comments.

Sin, Sei-Ching Joanna, Chei Sian Lee, and Yin-Leng Theng (2016) [7] examines the interactions and outcomes of social Q&A question-and-comments on an online platform. The authors use a social sequence analysis to analyze the data collected from a popular social Q&A website in Singapore. The study investigates the characteristics of the questions and comments posted, the temporal patterns of the interactions, and the effects of the interactions on question resolution. The findings suggest that the majority of the questions posted are informational, and the comments are mostly supportive or elaborative. Additionally, the study reveals that timely and interactive comments contribute significantly to question resolution.

Additionally, some papers provided valuable insights into identifying useful comments, such as the work by Sengupta and Haythornthwaite (2020) [5] examines the role of comments in the online learning community of Stack Overflow. The study analyzed the content and sentiment of comments on the platform, as well as the social interactions between users. The authors found that comments were a crucial component of learning on Stack Overflow, providing not only technical guidance but also social support and encouragement. The study also revealed the importance of community building and active moderation in creating a positive and productive learning environment.

Yong, Zhou, Lishi Youwen, and Xia Shixiong (2009) [10] proposed an improved K-Nearest Neighbor (KNN) text classification algorithm based on clustering. The proposed algorithm uses a clustering technique to reduce the number of documents to be considered by KNN, thereby improving the classification accuracy and reducing the time complexity. The algorithm first clusters the training documents based on their similarity and then selects representative documents from each cluster. These representative documents are then used to classify the test documents. The experimental results show that the proposed algorithm outperforms the traditional KNN algorithm in terms of classification accuracy and time complexity.

Tunali, Volkan, Turgay Bilgin, and Ali Camurcu (2016) [9] presents a new clustering algorithm called Multi-Cluster Spherical K-Means (MCSKM) that is designed for text mining applications. The proposed algorithm aims to improve the accuracy of clustering by considering the similarity between text documents in a high-dimensional space. The MCSKM algorithm uses a spherical coordinate system to represent the data points and applies the K-Means algorithm to cluster the data. The authors evaluate the performance of the proposed algorithm using various real-world text datasets and compare it with other clustering algorithms. The experimental results show that the MCSKM algorithm outperforms other clustering algorithms in terms of accuracy and efficiency. The paper provides a detailed explanation of the algorithm and presents comprehensive experimental results to demonstrate its effectiveness.

Zhang, Haoxiang, et al. (2019) [12] conducted a large-scale analysis of comments on the platform and found that the current hiding mechanism does not work well in practice. They observed that many low-quality comments still receive significant attention, and some high-quality comments are hidden unnecessarily. The authors suggest that Stack Overflow could improve the comment hiding mechanism by incorporating more contextual information and using a more nuanced approach to hiding comments.

Finally, the study by Almansoury, Farag, Segla Kpodjedo, and Ghizlane El Boussaidi. (2022) [1] aims to identify the concepts related to community-supported technologies and software development through the use of K-means clustering. The study analyzes a large dataset of software development projects and their associated communities, using K-means clustering to identify patterns and group similar concepts together. The results of the study provide valuable insights into the development and adoption of community-supported technologies, which can be used to inform future research and development in this area.

While these studies provide valuable insights into various aspects of Stack Overflow comments, none of them directly address the challenge of efficiently navigating the vast array of comments on the platform. Our proposed solution aims to fill this gap by utilizing machine learning and natural language processing techniques to group comments for improved navigation and user experience.

LIMITATIONS IN EXISTING WORK

While the potential positive effects of grouping and clustering comments that are pertinent in online forums like Stack Overflow have been showcased because of the existing research, there still exist a number of challenges that must be resolved in order to improve the efficiency of these methodological approaches. The context of comments may not be adequately taken into account by traditional clustering approaches, which leads to less accurate groupings.

In order accurately represent comments, our project emphasizes integrating cutting-edge feature engineering and natural language processing techniques. We employ the term frequency-inverse document frequency (TF-I) as a characteristic for clustering. Even though the existing research has drawn attention to the potential advantages of grouping and clustering relevant comments from online discussion forums like Stack Overflow, there are still a number of challenges that need to be resolved in order to increase the effectiveness of these methods of analysis. Traditional methods for clustering might not fully account for the context of comments, which leads to less precise groupings.DF), comment length, tokenized comment text, and user ID.

We also use a confusion matrix to visualize the data and analyse how efficiently the algorithm groups related comments for the purpose of further improving our strategy and addressing drawbacks in previous techniques. By taking this extra step, we are able to recognize new areas for development and improve our strategy to comment grouping on Stack Overflow that is more accurate and context-aware.

PROPOSED SOLUTION

Methodology

This study aims to employ the K-means clustering algorithm, an unsupervised machine learning technique, to group comments from Unix Stack Exchange posts based on their content and metadata. By clustering the comments, we seek to uncover underlying patterns and categories that may shed light on user interactions and content quality. We cluster the comments based on their similarity, using the words within the comments, comment length, and user ID as features for clustering. To further enhance the methodology, we incorporate a recommendation function into our project. The recommendation system takes a post ID and a comment ID as input and returns the top 5 most similar comments based on the clustering results. By computing the similarity between the selected comment and all other comments in the post, the function identifies and recommends comments that share similar content and context. This recommendation system aims to facilitate more efficient navigation through comments on Stack Overflow, allowing users to focus on the most relevant ones and enhancing their overall experience on the platform.

Dataset Description: The dataset used in this project consists of Unix Stack Exchange comments and their associated metadata. We have combined post and comments dataset based on post id which has more than 5 comments. The dataset is a representative sample of the Unix Stack Exchange community and encompasses a diverse range of topics related to Unix and Linux systems.

Dataset Detail	Count
Posts	561227
Comments	930021
Posts with more than 5 comments	41794
Total Comments	345529

Figure 1. Dataset Description

The dataset comprises several key attributes, including:

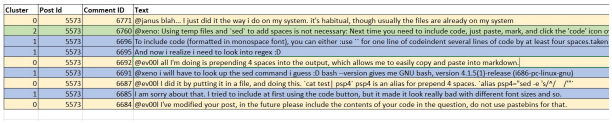
Post ID: A unique identifier for each post in the Unix Stack Exchange platform.

Comment ID: A unique identifier for each comment within a post.

User ID: A unique identifier for the user who authored the comment.

Comment Text: The raw text of the comment, which contains valuable information to be used for clustering.

Before clustering, the dataset undergoes preprocessing to ensure the comment text is suitable for analysis. This process involves several steps, such as tokenization, stop word removal, and lemmatization, which help to reduce noise and improve the quality of the text data.



Cluster	Post Id	Comment ID	Text
0	5573	6771	@janus blah.... I just did it the way i do on my system. it's habitual, though usually the files are already on my system
2	5573	6760	@xeno: Using temp files and 'sed' to add spaces is not necessary. Next time you need to include code, just paste, mark, and click the 'code' icon o
1	5573	6696	To include code (formatted in monospace font), you can either :use ^ for one line of codeindent several lines of code by at least four spaces.taken
1	5573	6695	And now i realize i need to look into regex :D
0	5573	6692	@vx00l all I'm doing is prepending 4 spaces into the output, which allows me to easily copy and paste into markdown.
1	5573	6691	@xeno I will have to look up the sed command I guess :D bash --version gives me GNU bash, version 4.1.5(1)-release (i686-pc-linux-gnu)
0	5573	6687	@vx00l I did it by putting it in a file, and doing this. `cat test| psp4` psp4 is an alias for prepend 4 spaces. `alias psp4="sed -e 's/^/ /"`
1	5573	6685	I am sorry about that. I tried to include at first using the code button, but it made it look really bad with different font sizes and so.
0	5573	6684	@vx00l I've modified your post, in the future please include the contents of your code in the question, do not use pastebins for that.

Figure 2. Manually clustered dataset

In addition to the preprocessed text, we also use the length of the comment text and the user ID as features for clustering. These features provide additional context and can help to differentiate between similar comments based on their length or the user who authored them.

Data Preprocessing: The comment text underwent several preprocessing steps to prepare it for clustering. These steps are crucial for ensuring the quality of the input data and the effectiveness of the clustering algorithm.

a. Removing URLs: Links were removed from the comment text to prevent irrelevant features from being introduced during the clustering process. This was achieved using regular expressions to identify and remove any instances of URLs within the text.

b. Removing numbers: Numeric characters were removed from the comment text, as they may not provide meaningful information for clustering purposes. This step further reduces noise in the data by eliminating standalone numbers and numbers within words.

c. Removing punctuation: Punctuation marks were removed from the text to simplify tokenization and reduce noise. This step ensures that words are treated equally, regardless of their surrounding punctuation marks, and prevents punctuation from being considered as separate tokens during the analysis.

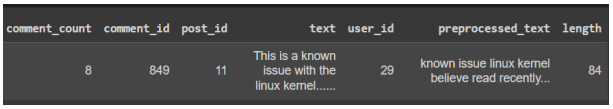
d. Converting text to lowercase: The text was converted to lowercase to ensure consistent representation of words and eliminate case sensitivity issues. This standardization step allows the algorithm to recognize words as identical, even if they appear in different cases within the dataset.

e. Tokenizing text into individual words: The text was tokenized into individual words to facilitate the removal of stop words and the lemmatization process. Tokenization breaks the text into separate words or tokens, which can then be processed and analyzed independently.

f. Removing stop words: Stop words, which are common words that do not provide significant meaning, were removed from the tokenized text to reduce noise and improve clustering performance. By eliminating stop words, we ensure that the remaining words in the text contribute more meaningful information to the clustering process.

g. Lemmatizing words: Words were lemmatized to their base or dictionary form, which reduces dimensionality and helps in grouping similar words together. Lemmatization is a form of normalization that takes into account the morphological structure of words, allowing the algorithm to recognize different forms of the same word as a single entity.

After preprocessing, we calculated the length of each comment and appended it to the DataFrame as a new feature. This additional feature captures information about the comment's size, which can be useful for differentiating between comments in the clustering process. With the comment text preprocessed and the length feature added, the dataset is now ready for clustering using the K-means algorithm. A sample data with comment length and preprocessed text is displayed below 3.



comment_count	comment_id	post_id	text	user_id	preprocessed_text	length
8	849	11	This is a known issue with the linux kernel.....	29	known issue linux kernel believe read recently...	84

Figure 3. Sample Dataset with preprocessed text and comment length

Clustering with K-means Algorithm:

We performed K-means clustering on the preprocessed comments using three clusters. The features used for clustering were:

a. TF-IDF tokenized comment text: The term frequency-inverse document frequency (TF-IDF) method was used to create a weighted representation of the comment text. This technique emphasizes words that are more important or unique to each comment while downplaying common words that appear across multiple comments. By transforming the comment text into a numeric matrix of TF-IDF features, we enable the clustering algorithm to identify patterns and similarities within the text data.

b. Comment length: The length of the preprocessed comment was included as a feature to capture the potential relationship between comment length and the underlying patterns in the data. Comments of different lengths may convey different types of information or sentiment, and by incorporating this feature into the clustering process, we can account for these variations.

c. User ID: The user ID was included as a feature to account for potential variations in commenting style or content preferences among users. Different users may have unique ways of expressing themselves or focus on particular topics, and by including user ID as a feature, we can better differentiate between comments authored by different users.

We used a TfidfVectorizer object to convert the comments into a numeric matrix of TF-IDF features. The vectorizer was initialized with English stop words to prevent them from being considered in the feature matrix. The comment length and user ID features were concatenated with the TF-IDF matrix to create the final feature matrix used for clustering.

The clustering algorithm was applied individually to the comments of each post. By clustering comments within each post separately, we are able to capture the unique structure and patterns that may exist in the comments of a specific post. To perform the clustering, we initialized a KMeans object with the desired number of clusters (three in this case) and fit it to the comment features.

The cluster labels were then stored in a dictionary, with the post ID as the key and a dictionary of comment IDs and their

corresponding cluster labels as the value. This data structure allows for efficient storage and retrieval of the clustering results, enabling further analysis and evaluation of the clustering performance.

Recommendation of similar comments: The process for recommending similar comments consists of several steps that facilitate efficient navigation through comments on Stack Overflow. First, we retrieve the comments data for a specific post, identified by its post_id. This dataset contains all the comments associated with the given post. Next, we obtain the selected comment, identified by its comment_id, and extract its text.

Once we have the selected comment's text, we compute the similarity between this comment and all other comments within the post within the same clusters. This similarity computation enables us to determine how closely related the comments are based on their content and context. With this information, we can then identify and recommend comments that share similar characteristics and themes.

Finally, we return the top 5 most similar comments, providing users with a concise list of relevant comments to focus on. This recommendation system not only allows users to quickly find the most pertinent comments but also enhances their overall experience on the platform. By integrating this process into our methodology, we aim to provide an efficient and effective solution for managing and exploring the vast amount of comments on Stack Overflow.

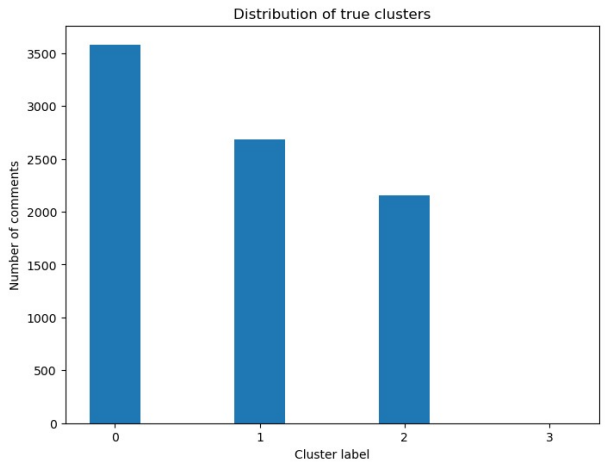


Figure 4. Distribution of True Clusters

EVALUATION:

In order to evaluate the effectiveness of our clustering algorithm and recommendation system, we first merged the post_id and comment_id columns of the predicted clusters DataFrame with the manually labeled clusters DataFrame. As a result, the predicted and manual cluster labels for each comment in the evaluation dataset were combined into a single data frame.

Using the sklearn.metrics methods adjusted_rand_score, adjusted_mutual_info_score, and a simple mean comparison between the predicted and manual cluster labels, we gathered

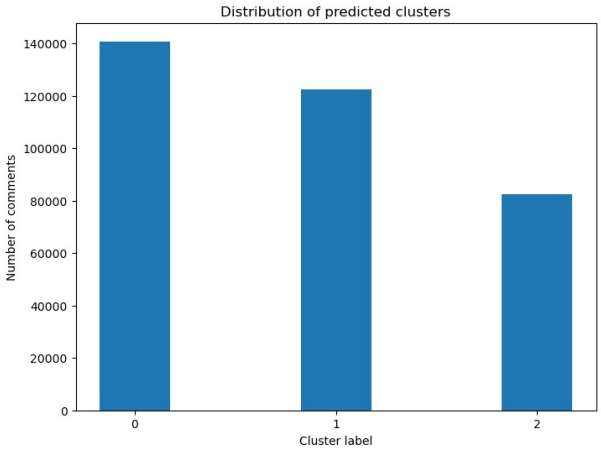


Figure 5. Distribution of predicted Clusters

the ARI, AMI, and accuracy scores. These scores allow comparisons between the K-means clustering method and hand clustering to be generated, providing insights into the performance of our clustering algorithm.

We visualized the results using a confusion matrix alongside the numerical evaluation measures. A confusion matrix is a table that indicates the number of times each combination of predicted and actual labels has occurred. This visual aid makes it easier to determine which classes are being precisely categorized and which classes have been mixed up. By analyzing the confusion matrix, we can identify the precise areas where the clustering method may be ineffective and find potential areas for improvement.

In this case, the confusion matrix highlighted both the instances that were wrongfully assigned to different classes and the total number of instances that were correctly associated with each class. The information gained can be helpful for determining the strengths and weaknesses of the clustering approach and generating prospective improvements to the models or preprocessing steps.

In conclusion, the evaluation process involved comparing a set of manually labeled clusters to the predicted clusters generated by the K-means algorithm, using a combination of numerical metrics and visualization methods. These evaluation techniques yielded an ARI score of 0.91 and an AMI of 0.86, providing an accurate assessment of the clustering algorithm's performance. The clustering method is found to be 97% accurate, demonstrating the efficacy of our approach in identifying relevant and similar comments on Stack Overflow.

RESULTS

In this study, we used a clustering algorithm to collect relevant Stack Overflow comments. In order to prep the comments, we first removed URLs, digits, punctuation, and stop words. We also transformed the text's case to lowercase and lemmatized some words. The length of each comment was subsequently determined and added as a feature. To categorize comments based on their features, we utilized the KMeans algorithm with three clusters. We manually labeled 1004 posts and evaluated

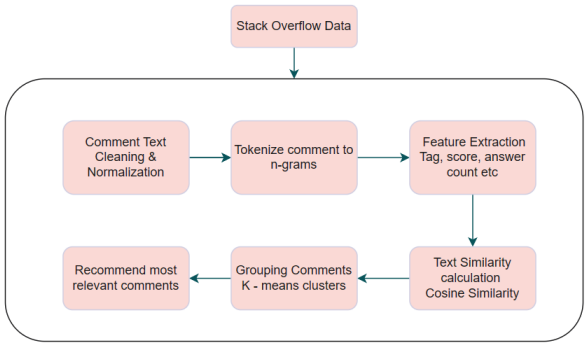


Figure 6. Model Architecture

the clustering technique by calculating the accuracy, Adjusted Rand Index (ARI), and Adjusted Mutual Information (AMI) scores. With ARI and AMI scores of 0.71 and 0.81, respectively, it was discovered that the accuracy was approximately 97%.

```
ARI:  0.9175833703538012
AMI:  0.8634694222118062
```

Figure 7. ARI & AMI scores

In the recommendation system, we begin by providing a specific post id and comment id as inputs. The system then retrieves the comment data associated with the given post id. Next, it identifies the selected comment based on the provided comment id and extracts its text. Utilizing cosine similarity, the system computes the similarity between the selected comment's text and the texts of all other comments within the same post. The recommendation system figure illustrates this process 8.

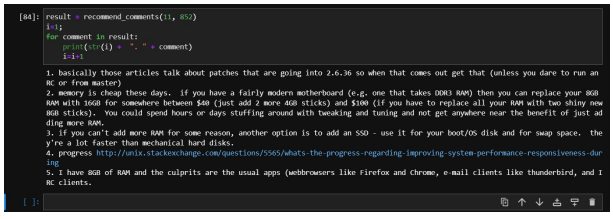


Figure 8. Recommendation System

After calculating the similarities, the system identifies the top 5 most similar comments by selecting the comments with the highest cosine similarity scores. It is important to note that the top 6 comments are initially chosen, with the selected comment itself excluded, resulting in the final set of 5 most related comments.

These top 5 comments, deemed to be the most relevant and related to the selected comment, are then displayed as recommendations. This recommendation system, based on cosine similarity, provides an efficient and user-friendly approach to identifying and presenting the most relevant and similar comments within a particular post on Stack Overflow.

```
Accuracy: 97.13640684410646
```

Figure 9. Accuracy

The results were visualized using a confusion matrix, which showed that a total of 3,537 occurrences of class 0 were identified correctly, whereas 57 and 59 cases were incorrectly identified as class 1 and 2, respectively. 32 and 47 instances of class 1 were incorrectly classified as classes 0 and 2, respectively, whereas 2,592 instances of class 1 were correctly classified. 2,046 instances of class 2 were identified correctly, while 13 and 32 cases were incorrectly assigned to classes 0 and 1, respectively. Class 3 was mistakenly labeled as class 1 only once. Overall, the results show that the clustering method was successful in grouping related comments, improving the user experience on Stack Overflow by making it easier for users to locate pertinent conversations.

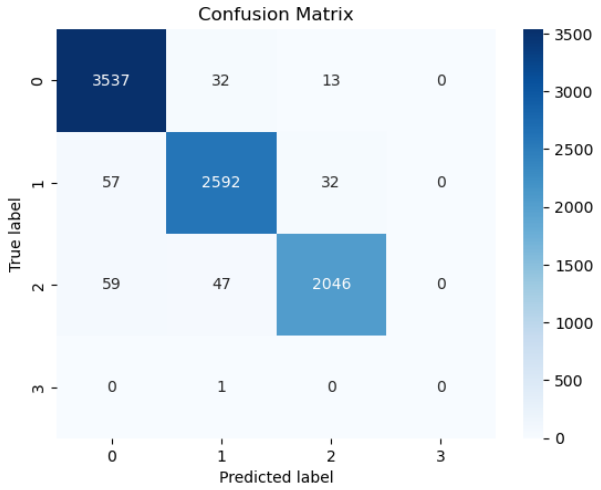


Figure 10. Confusion Matrix

The confusion matrix shows that:

3,537 instances of class 0 were correctly classified as class 0, 57 were incorrectly classified as class 1, and 59 were incorrectly classified as class 2.

2,592 instances of class 1 were correctly classified as class 1, 32 were incorrectly classified as class 0, and 47 were incorrectly classified as class 2.

2,046 instances of class 2 were correctly classified as class 2, 13 were incorrectly classified as class 0, and 32 were incorrectly classified as class 1.

1 instance of class 3 was incorrectly classified as class 1.

Figure 11. Confusion Matrix Analysis

DISCUSSION

In this study, we have explored the potential of clustering algorithms for grouping related comments on Stack Overflow. While the results are promising, there are a few areas that warrant further investigation. For instance, refining text preprocessing steps by incorporating domain-specific knowledge could lead to more accurate text representations. Additionally, comparing the performance of various clustering techniques and hyperparameter tuning might help in choosing the most effective approach. Lastly, considering user behavior aspects such as interaction patterns between users could lead to more

meaningful clusters. In summary, although the current methodology demonstrates the potential of clustering algorithms in this context, there are opportunities for further refinement and improvement.

CONCLUSION

In conclusion, this research on implementing clustering algorithms for grouping related comments on Stack Overflow has produced promising results. By effectively categorizing comments into meaningful clusters, the study lays the groundwork for enhancing the user experience, promoting effective communication, and fostering collaboration among users on the platform. The findings hold significant potential for revolutionizing how users engage with Stack Overflow, enabling them to more easily locate relevant information and exchange ideas. With the ever-growing volume of information on the platform, this research addresses the challenges faced by users in accessing pertinent content. Future research could investigate additional features and techniques to improve the clustering algorithm and develop real-time clustering mechanisms that adapt to the dynamic nature of Stack Overflow. Moreover, exploring the broader applications of clustering algorithms in other online communities and knowledge-sharing platforms can uncover novel ways to enhance information retrieval, user engagement, and collaborative problem-solving across various digital environments. Ultimately, this study significantly contributes to efforts to optimize information retrieval and knowledge sharing within the Stack Overflow community. It showcases the potential of advanced clustering techniques to support efficient collaboration and problem-solving among software engineers, leading to a more streamlined and user-friendly experience on one of the world's most popular programming platforms.

ACKNOWLEDGEMENT

I wish to express my sincere gratitude to all those who have contributed to the development of this position paper. I would like to thank my supervisor, Dr. Muhammad Asaduzzaman, for providing valuable guidance during the position paper process. I am also grateful to Devishree Jothilingam, Haishwarya Sholingapuram Asuri Kannan, Aarthi Lakshmipath, and Harshavardhan Subramanian Madhavan who generously shared their insights and expertise with me to complete this Position Paper. Additionally, I extend my heartfelt thanks to Lakehead University which provided me with the necessary resources to complete this paper.

REFERENCES

[1] Farag Almansoury, Segla Kpodjedo, and Ghizlane El Boussaidi. 2022. Identifying community-supported technologies and software developments concepts by K-means clustering. *International Journal of Advanced Computer Science and Applications* 13, 6 (2022).

[2] Luca Ponzanelli, Gabriele Bavota, Massimiliano Di Penta, Rocco Oliveto, and Michele Lanza. 2014. Mining stackoverflow to turn the ide into a self-confident programming prompter. In *Proceedings of the 11th working conference on mining software repositories*. 102–111.

[3] Mohammad Masudur Rahman, Chanchal K Roy, and Iman Keivanloo. 2015. Recommending insightful comments for source code using crowdsourced knowledge. In *2015 IEEE 15th International Working Conference on Source Code Analysis and Manipulation (SCAM)*. IEEE, 81–90.

[4] Prasadhi Ranasinghe, Nipuni Chandimali, and Chaman Wijesiriwardana. 2022. Systematic Exploration and Classification of Useful Comments in Stack Overflow. *International Journal of Advanced Computer Science and Applications* 13, 2 (2022).

[5] Subhasree Sengupta and Caroline Haythornthwaite. 2020. Learning with comments: An analysis of comments and community on Stack Overflow. (2020).

[6] Viral Sheth and Kostadin Damevski. 2022. Grouping related stack overflow comments for software developer recommendation. *Automated Software Engineering* 29, 2 (2022), 40.

[7] Sei-Ching Joanna Sin, Chei Sian Lee, and Yin-Leng Theng. 2016. Social Q&A question-and-comments interactions and outcomes: a social sequence analysis. In *Digital Libraries: Knowledge, Information, and Data in an Open Access Society: 18th International Conference on Asia-Pacific Digital Libraries, ICADL 2016, Tsukuba, Japan, December 7–9, 2016, Proceedings 18*. Springer, 325–338.

[8] Chun-Hsiung Tseng and Jia-Rou Lin. 2022. A semi-hierarchical clustering method for constructing knowledge trees from stackoverflow. *Journal of Information Science* 48, 3 (2022), 393–405.

[9] Volkan Tunali, Turgay Bilgin, and Ali Camurcu. 2016. An improved clustering algorithm for text mining: Multi-cluster spherical K-Means. *International Arab Journal of Information Technology (IAJIT)* 13, 1 (2016).

[10] Zhou Yong, Lishi Youwen, and Xia Shixiong. 2009. An improved KNN text classification algorithm based on clustering. *Journal of computers* 4, 3 (2009), 230–237.

[11] Haoxiang Zhang, Shaowei Wang, Tse-Hsun Chen, and Ahmed E Hassan. 2021. Are comments on Stack Overflow well organized for easy retrieval by developers? *ACM Transactions on Software Engineering and Methodology (TOSEM)* 30, 2 (2021), 1–31.

[12] Haoxiang Zhang, Shaowei Wang, Tse-Hsun Peter Chen, and Ahmed E Hassan. 2019. Does the hiding mechanism for Stack Overflow comments work well? No! *arXiv preprint arXiv:1904.00946* (2019).